Database project

Case Study

Circuit Surge eSports is a professional gaming organization based in London. They have several teams across different gaming titles such as League of Legends, Fortnite, Valorant and StarCraft. The organization is responsible for the training, performance monitoring, strategy development, competitions scheduling, fan interaction, and promotional activities of these teams. The competitions are organised in UK and international alike.

Currently, each of these facets of their operations are managed via different tools, and there is no centralized system for coordinating these aspects. Fans, sponsors, and stakeholders don't have a unified platform to interact with the teams or access real-time updates and stats.

Circuit Surge eSports is looking for a comprehensive software solution that can cater to their unique needs. The software should include:

- Training module: Tracks the training schedules, skill progression, and fitness routines of players.
- Performance module: Monitors player's game stats, game win/loss ratios, individual game performance, etc.
- Strategy module: Stores information about different game strategies, team compositions, opponents' weaknesses and strengths, etc.
- Competition module: Maintains the schedule of upcoming tournaments, matches, results, prize earnings, and rankings.
- Fan Interaction module: Handles fan meet schedules, fan queries, merchandise sales, and fan voting for favourite player/strategy.
- Promotion module: Tracks social media posts, promotional campaigns, sponsorships, and collaborations.

Each team and player should have unique profiles linked to these modules. A player can be only in one team, and a game might have multiple teams. Each module will generate documents that will be uploaded as PDFs, and the software should have the capability to track these documents via unique IDs. The PDF will be created by a web application based on selected criteria, hence you do not need to worry about creating/storing PDFs but to be able to track them.

In addition, the software needs to manage three crucial aspects:

- Roster Management: Tracks the contract status of the players (e.g., contract start/end date, contract terms), recruitment of new players, and player transfers.

- Game Updates: Records any updates or patches in the games that the teams are playing and how they affect the game strategies.
- Incident Reporting: Registers any incident that impacts the team or its players (e.g., player injuries, violation of game rules, technical issues during games). The incident reporting should include the date/time of occurrence, the individuals involved, and the person who reported the incident.

Please read carefully and submit ALL required elements.

The initial design was for the single HQ location. However, the company is looking for new solutions as they are planning to expand in the future with 4 more branches in different locations. The current solution of a single relational database will no longer be suitable from a cost and efficiency point of view. You have appointed to provide a report that will cover the following aspects:

2.1. Critically analyse options to the architecture of database systems; they may be Distributed Databases, Web Databases, NoSQL Databases or Mobile Databases that might be suitable for the client. The discussion should not be descriptive (as what is a DD) but an analysis if it is suitable for the business needs (or not) with arguments.

(**20 points**)

2.2. Based on the analysis conducted (2.1) recommend one of the suitable architectures. The recommendation should include the additional design requirements, including, but not limited to, data allocation, data fragmentation, data replication, optimization, distributed backup and recovery strategies, distributed query processing and underlying infrastructure changes from the centralised system discussed in Project1. Chose only one of the following:

a. Distributed Database System

b. NoSQL Database

c. Mobile Database

NB: *This aspect is NOT about mobile databases that reside on phones (mini databases – SQLite), it is about mobile access to a centralised database. It is about what data will be stored/replicated on the mobile device in order to access/sync with the main database.*

*(**50 points**)*

2.3. Following up the architecture decision (2.2) critically discuss and propose some suitable security measures that should be considered for the new proposed solution. What would be the technical implications, flaws, risks, advantages etc. The discussion should be relevant to the proposed architecture and case study and not to be generalised as something that can be applied to any database.

(**20 points**)

2.4. Innovation and excellence - Awarded for extraordinary work that goes beyond minimum scenario description and requirements, including but not limited to organised layout, grammar, references and anything else that is outstanding.

(**10 points**)

No	Element	Description	Max points
Project 2			
1	DB Alternatives	- A critical analysis and discussion of which architecture would be suitable for migrating from a centralised database	**20**
2	DB Distribution Solution	- Based on your analysis propose (design) the changes for the alternatives - Chose only one alternative - only some of the below will apply dependent on the chosen architecture - Designs (e.g. data fragmentation, replication, location) - JSON structure - Physical partitions, key-values, graph, document etc. - Anything else that is relevant to the architecture design	**50**
3	Security Considerations	- Based on the distribution design evaluate some of the security measures that should be put in place - The measures should be relevant to the case study and proposed solution and not general	**20**
6	Innovation & Excellence	- Awarded for extraordinary work that goes beyond minimum scenario description and requirements, including but not limited to organised layout, grammar, references and anything else that is outstanding	**10**
Please READ notes.		**TOTAL points 100**	

Note[2]**:** *You don’t need to create any code or to physically implement the proposed design but plan and critically evaluate.*

Points Distribution	
0%	0 is reserved for non-submissions or plagiarism
1% - 19%	- No attempt to cover the requirement, very poor scholarship, incomplete project etc
20% - 39%	- Poor attempt at covering project, no understanding of the database concepts; the proposed solution is ambiguous with different elements from different architectures etc.
40% - 49%	- A minimal discussion about DB alternatives; the proposed solution is very general without being referred to the case study etc